RESEARCH PAPER

Integrated "Generate, Make, and Test" for Formulated Products Using Knowledge Graphs

Sagar Sunkle, Deepak Jain, Krati Saxena†, Ashwini Patil, Tushita Singh, Beena Rai & Vinay Kulkarni

Tata Consultancy Services Research Pune, Maharashtra, India, 411013

Keywords: Formulated product design; Formulation recommendation; Formulation template; Robotic labs; In-silico testing; Integrated generate-make-test

Citation: Sunkle, S., et al.: Integrated "generate, make, and test" for formulated products using knowledge graphs. Data Intelligence 3(3), 340-375 (2021). doi: 10.1162/dint_a_00096
Received: December 31, 2020; Revised: April 5, 2021; Accepted: April 30, 2021

ABSTRACT

In the multi-billion dollar formulated product industry, state of the art continues to rely heavily on experts during the "*generate*, *make* and *test*" steps of formulation design. We propose automation aids to each step with a knowledge graph of relevant information as the central artifact. The *generate* step usually focuses on coming up with new recipes for intended formulation. We propose to aid the experts who generally carry out this step manually by providing a recommendation system and a templating system on top of the knowledge graph. Using the former, the expert can create a recipe from scratch using historical formulations and related data. With the latter, the expert starts with a recipe template created by our system and substitutes the requisite constituents to form a recipe. In the current state of practice, the three steps mentioned above operate in a fragmented manner wherein observations from one step do not aid other steps in a streamlined manner. Instead of manually operated labs for the *make* and *test* steps, we assume automated or robotic labs and in-silico testing, respectively. Using two formulations, namely face cream and an exterior coating, we show how the knowledge graph may help integrate and streamline the communication between the *generate*, the *make*, and the *test* steps. Our initial exploration shows considerable promise.

† Corresponding author: Krati Saxena (Email: krati.saxena@tcs.com; ORCID: 0000-0001-7049-9685).

1. INTRODUCTION

We encounter formulated products many times in our daily lives. Products like personal, home, industrial care, pharma and health care, coatings (paints) and surfaces (lubricants, adhesives), and confectionary foods and drinks are pervasive in their use. The formulated products industry is an expanding global market of around 1400B Euro [1]. Despite this scale, the state of the art in designing formulated products relies heavily on the experts' experiential knowledge.

The design of formulated products involves distinct steps. First, the expert needs to arrive at a feasible recipe for the product. This step involves searching and selecting ingredients, weights, possible mixtures, and recipe steps containing process actions at certain conditions. Experts carry out the activities manually, i.e., searching ingredients with specific functionalities, combining them with other ingredients, and deciding upon what is done to them. The following two steps involve making the product and testing it for its intended purpose and consist of considerable experimentation. Overall, significant manual intervention at every step leads to considerable time to market, many times in months and years, and enormous cost. These steps are individually carried out in silo [2, 3, 4, 5, 6, 7]. It is possible to imagine digitalization and automation in each of the steps mentioned above. Reliance on experts and the siloed nature of formulations design means they do not fully benefit from automation in any steps.

We propose aided formulation recipe generation by storing information relevant to formulations as a knowledge graph and creating recommendation and template generation and substitution systems on top of this knowledge graph. Additionally, we show that if formulation making and formulation testing had automated or in-silico realizations, it is possible to use the knowledge graph as the connecting link between the three steps, reducing the siloed nature and benefiting from observations in each step. Our specific contributions are as follows:

We aid the expert in generating formulation recipes in two different ways:

- The expert may design the formulation from scratch, receiving recommendations using the knowledge graph on ingredients, mixtures, and weights, actions to be applied to ingredients, and conditions at which to apply the actions [8].
- The expert may start with a templatized recipe of the intended formulated product. We show how such a template can be created using the knowledge graph. In this case, the expert substitutes representative ingredients, also in aided manner, to achieve the same result.

Following the *generate* step, as stated above, we expect the *make* and *test* steps to take place using automated robotic labs and in-silico manner, respectively. With two examples, a cosmetic formulated product and a paint formulated product, we show how to use the knowledge graph to streamline sharing of information among the three steps.

As illustrated in Figure 1, using a knowledge graph as a central connecting artifact between aided formulation generation, automated formulation making, and in-silico testing, we aim to reduce the over-reliance on experts and help integrate the largely siloed steps in formulation design.

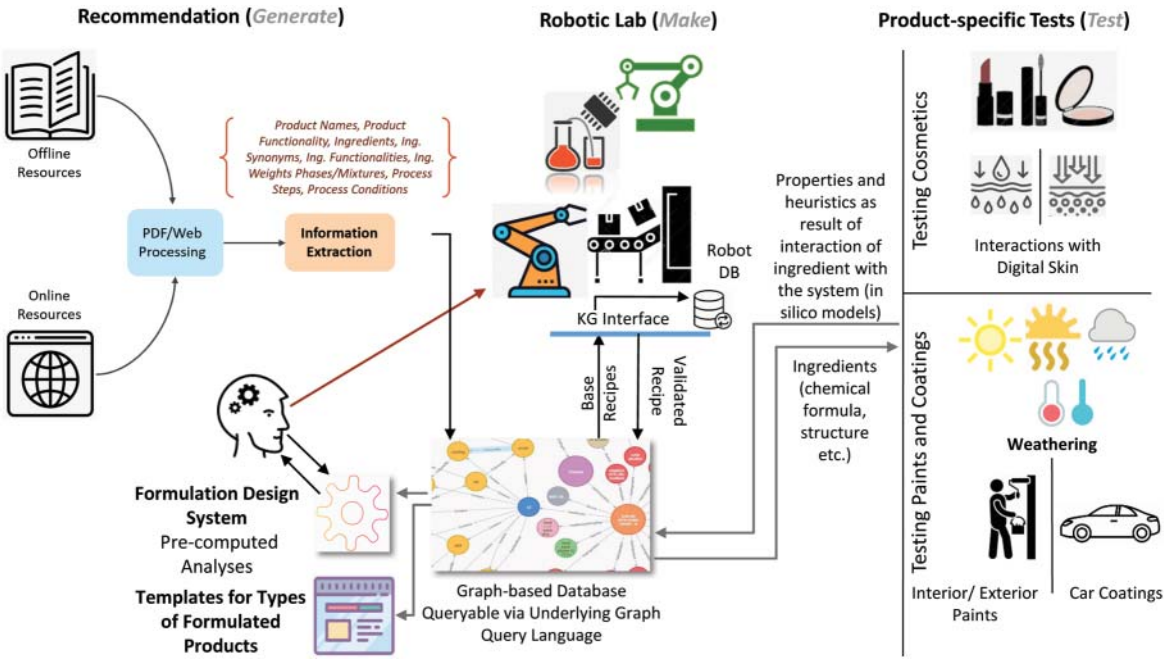


Figure 1. Integrated generate, make, and test for formulated products.

Of the two ways to generate formulations, namely starting from scratch with the intended product and starting with a template of the intended product, we covered the former in [8]. In Section 3, we recount it to contrast it with the latter. Both ways require the extraction of several different kinds of information and storage as a knowledge graph, as shown in Figure 1 and various analyses to be performed before proceeding with the generation of recommendations.

We assume interfacing systems in both the robotic lab and in-silico testing to exchange information with the knowledge graph. The *generate* step provides the base recipe to the robotic lab system representing the *make* step and receives a validated recipe in return, as shown in Figure 1. We explain this in detail in Section 4. Similarly, in-silico testing provides information about tests conducted in the form of physiochemical attributes of ingredients and mixtures to the *generate* step to store in the form of heuristics. These heuristics can be queried in the future when iteratively conducting the *generate*, *make*, and *test* steps. In Section 4, we show two different formulated products: face cream and an exterior coating carried through these steps in the manner described above. Section 5 concludes the paper. We begin next with related work.

2. RELATED WORK

2.1 Formulated Products

A formulated product is a product with well-defined target properties and contains a minimum of two selected, processed, and combined ingredients [4, 9, 10, 11, 12]. The word *formulation* is referred to by many senses of the word; for instance, it may mean a *recipe*, i.e., a list of ingredients with processing steps. It also denotes the *act of formulating*, meaning the combination of processes used for mixing and conditioning of active, protective, or stabilizing ingredients and the know-how that enables the selection of ingredients. Finally, it also indicates the *actual blend of ingredients*.

Their ubiquitous usage in our daily lives makes the formulated products industry a vast enterprise with a multi-billion-dollar turnover[①] each year. Chemical products go through design and development through mainly manual heuristic-rule-based, trial-and-error experiment-based approaches [5, 6, 7]. As a subset, this applies to formulated products as well. These products contain ingredients that undergo a step-by-step procedure that may include heating, cooling, stirring, and mixing to obtain specific target properties, both physical and chemical [13].

Individual ingredients may provide active functionality, as in active pharmaceutical ingredients in medicine, and enable enhanced delivery (especially in skin-applicable formulated products such as many cosmetic products or pharmaceutical pastes) or as a protective or stabilizing agent [9, 14, 15].

Many ingredients used in formulated products are multi-functional [15, 16]; for instance, Cetyl Alcohol is used as an emulsifier and a thickening agent in cosmetic products. In contrast, as a food additive, it is used as a flavoring agent. Even though an ingredient may be multi-functional, it is generally used for its primary functionality for specific formulated products. The enormous number of ingredients available to make various kinds of formulated products pose the following concerns [10, 12, 15, 16]: If a specific functionality such as an emulsifier is needed, which representative ingredient to use?; Should the ingredients be processed separately or as parts of a mixture (generally referred to as phase in formulation texts); How will the choice of ingredients and relative quantities affect the properties of the product, esp. sensory attributes?; What steps to follow in what order to arrive at the final product?. Such complexity has led many researchers to propose formulation design frameworks which we review next.

2.2 Formulation Design Frameworks

Many authors have come up with formulation design frameworks. A generic framework for chemical product design [10], also applicable to formulated products, starts with the identification of consumer needs [17], translating these needs to chemical/physical properties [2, 18, 19, 20], to manufacturing that product [3, 11, 21]. Design frameworks focusing on formulated product design discuss all three steps in formulated product design, namely *generate*, *make*, and *test* [3, 5, 15, 17, 20, 21, 11] but often without considering automated or robotic labs for the *make* step and without in-silico realization of the *test* step.

① EU Formulation Network: Formulated Products at https://formulation-network.eu/about/objectives

Nearly all design frameworks discuss process design with equipment and their operating conditions and optimal economic parameters.

Based on our interactions concerning digitalization and automation with some of the largest formulated product companies, we feel the need to consider automated make and in-silico test steps in concert with the aided generate step in formulated product design.

2.3 Automated "Make"

Automated laboratories refer to labs where robots carry out various tasks right from picking up specific ingredients as per the recipe, from a particular shelf/container to measuring the correct quantity of the material to be added to performing the appropriate action (mixing, cooling, heating, etc.)

The concept of high-throughput (HT) screening has been around for more than five decades [22, 23, 24] and has been applied across wide range of domains like electrolytes [25], catalysts [26], and biomedical research [27].

Most recently, there have been specific mentions to high throughput formulation engines fully automated robotized systems which can make and test 100s of formulations per day. Evonik's High Throughput Equipment (HTE)[②] is 2 meters high, occupies 120 square-meter of area, has 13 robots performing various tasks, and can churn out 120 formulations on an average in 24 hours.

On the other hand, Materials Innovation Factory has recently launched a bespoke facility, viz. Formulation Engine[③] with an investment of around 3 million pounds to enable entirely automated making and testing of formulations. It is a modular facility allowing up to 6 separate processing and testing stations to be connected.

The concept of *autonomous* or self-driving laboratories is comparatively a more recent concept, where automation exists in conjunction with artificial intelligence (AI). AI suggests what experiments to perform next based on past learning experiences and robotic platforms merely follow instructions. Various research groups have explored this concept for accelerating scientific discovery in a myriad of applications like battery electrolytes [28], thin-film materials [29, 30], inorganic compounds [31, 32], scientific instruments [33], clean energy [34], natural products [35], and alloys [36].

ChemOS is another recent attempt for a generic architecture for autonomous laboratories [37]. It consists of orchestration software with fundamental layers of database management, experiment scheduling, and designing, feedback, etc. which can coordinate the overall workflow in a typical autonomous lab.

Gromski et al. [38], and Steiner et al. [39], described an abstraction called *Chemputer*, which mirrors the working of a chemist and enables linkage to physical operations of the automated robotic platform. They also described a program called *Chempiler* to produce machine-level instructions for the synthesis robot.

② EVONIK-HTE https://bit.ly/3purvBM
③ Formulation Engine: Materials Innovation Factory https://bit.ly/37RiD3r

A common takeaway is that most of these works essentially solve (or have modules that solve) the experiment design problem by posing it as an optimization problem. The objective function usually describes deviation in the actual and desired property of material/formulation under consideration. The algorithm searches for the most optimum set of inputs (weight fractions, volumes, process parameters, etc.), which minimize the objective function, in turn recommending the next instance of the experiment to be performed. This contrasts with manually operated experiments, that include a factorial number of (design of) experiments (DOEs), thereby consuming considerably more resources and time.

2.4 In-silico "Test"

Testing is one of the most crucial steps in realizing a formulated product as it decides whether the formulation goes to production. Testing aims at such purposes as evaluating the product against the customer brief and for compliance/registration[④]. Testing is conducted at various stages across the formulated product design lifecycle- (a) at the ingredient level and (b) at the product (formulation) level and may result in changes (either in choice of ingredient or process parameters, etc.). The tests conducted can be categorized broadly into tests for evaluating (a) physiochemical properties such as density, viscosity, pH levels, conductivity, and surface tension, (b) safety properties such as toxicity and flammability, (c) efficacy properties like release profile and bioavailability, (d) stability properties such as phase separation and interaction with the environment, and finally but most importantly e) product brief related properties.

All these tests are performed using specific instruments[⑤] by following the specific procedure, which may differ as per standards being followed, such as ASTM[⑥] and ISO[⑦], which detail the experimental setup and the conditions for the tests.

Although most of these tests are conducted manually (or by robots in the recent attempts of autonomous labs), there have been various attempts at either coming up with empirical relations, mathematical models, or exploring modelling and simulation (based on first principles, multi-scale) to reduce the time and resources utilized during some of these tests.

Our digital skin platform[⑧] uses multi-scale modelling and virtual reality (VR) for transdermal pharmaceutical and cosmetics delivery [40]. The platform leverages micro-and macro-scale modelling [41] and is capable of in-silico testing [42]. It can emulate the physicochemical properties of human skin, thereby facilitating the study of how constituents of new formulations are transported through its layers [43].

We believe that the best way to enable the integration of *generate*, *make*, and *test* activities is to store the knowledge required to design formulated products and share it across the three steps. We review the different kinds of information of interest for this next.

④ EU Regulation for Testing of Cosmetic Products https://bit.ly/3mPGggP
⑤ Product Testing Instruments https://www.anton-paar.com/in-en/products/
⑥ American Society for Testing and Materials https://www.astm.org/
⑦ International Organization for Standardization https://www.iso.org/standards.html
⑧ The TCS Digital Skin Twin https://www.tcs.com/tcs-digital-skin-twin-platform

2.5 Kinds of Knowledge in Formulated Products Industry

Several researchers discuss the information that can be stored as knowledge to design formulated products effectively [5, 12, 15, 16, 20]. Figure 2 shows various pieces of information from the domain of formulated product design to store and utilize as knowledge.

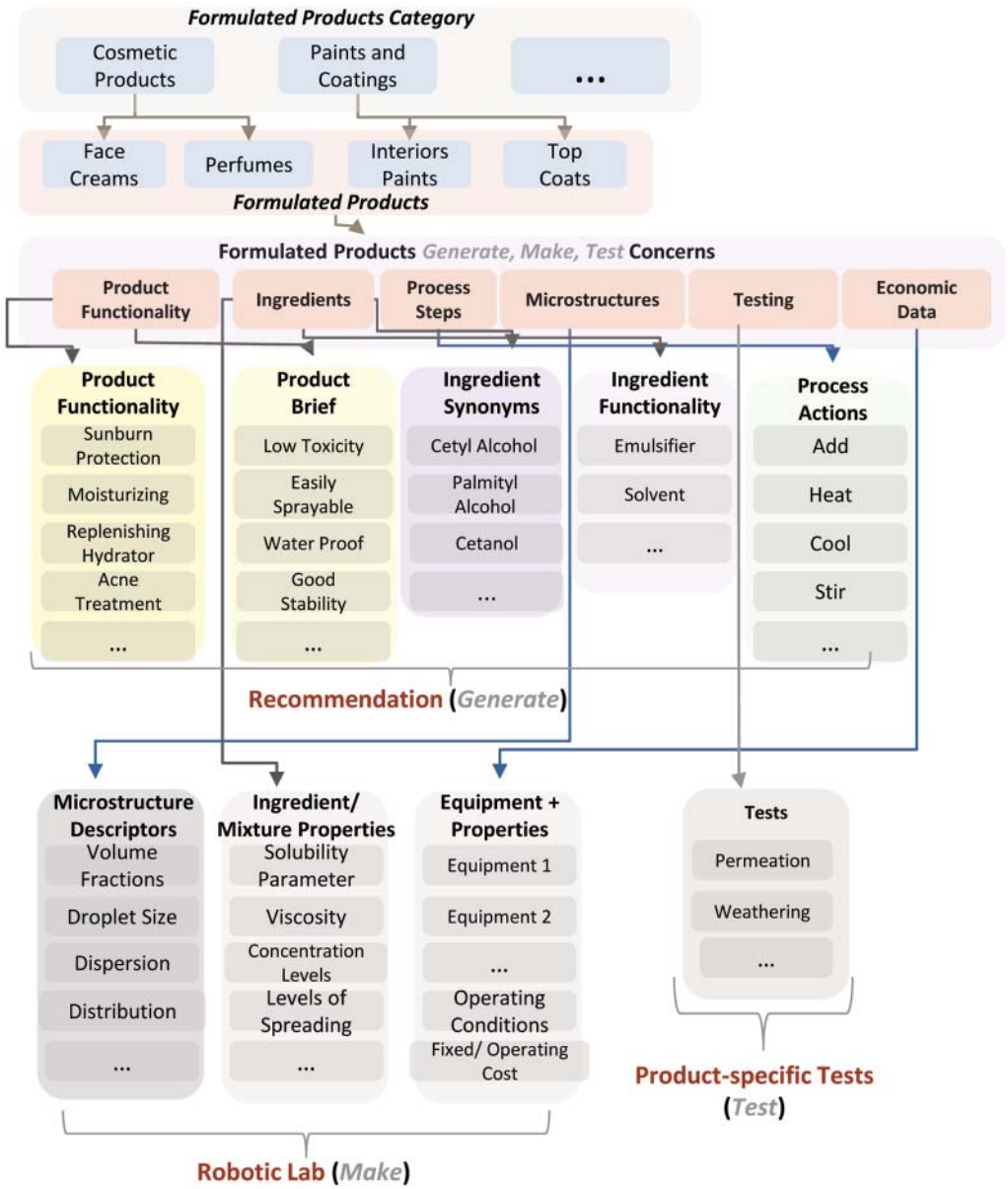


Figure 2. Kinds of knowledge required for formulated products design.

Depending on the type of formulated products, such as cosmetic products or paints and coatings, several specific products exist within. For instance, cosmetic products may be antiperspirants and deodorants, baby products, bath and shower products, beauty aids, creams, fragrances and perfumes, hair care products, insect repellents, lotions, shampoos, shaving products, soaps, sun care products, and so on [44]. Similarly, paint products could be coatings and topcoats, coil coatings, enamels, exterior and interior paints, lacquers, primers, sealers, stains, texture paints, etc. [45].

Individually each of these could be considered a category in itself. For instance, creams could be cleansing cream, cold cream, night cream, anti-inflammatory cream, anti-wrinkle cream, and so on. *Product functionality* captures the specific purposes of the product. It is possible to carry out analyses at the required level by storing the details of formulation types and categories.

For individual products, ingredients assume the most vital role. Since individual recipes may contain references to various *synonyms* of ingredients, storing this information is necessary. The next critical piece of information is *ingredient functionality*. Formulations are the combination of functionalities offered by the ingredients used; for instance, creams, in general, tend to contain an emulsifier, an emollient, and a thickener. Process steps are the actions applied to ingredients in standalone or as phases/mixture to obtain the target product. We discuss *ingredient synonyms*, *functionalities* as well as *process steps* later in Section 3.

The *ingredient/mixture properties*, as well as *microstructure descriptors*, describe physical properties. *Equipment* and *properties of equipment*, along with *operating conditions* and *economic data* on fixed and operational costs, are referred to in the make step. We discuss some of these later in Section 4.

Product brief could be considered a qualitative function of the ingredients, often describing consumer preferences with sensory and other fuzzy properties [20]. These are often mapped to a combination of *ingredient functionalities*, *ingredient/ mixture properties*, and *microstructure descriptors* and treated as heuristics to arrive at a suitable qualitative requirement [15, 20, 46, 47].

In the next section, we show how we process and analyze various pieces of information shown in Figure 2. Storing this information as a knowledge graph enables recommending ingredients, weights, process actions, and conditions to the expert as he or she generates a formulated product candidate. It is also possible to generate a template for a specific formulated product. We discuss both these ways of *generate* step in the following.

3. KNOWLEDGE GRAPHS

We discussed how we extract various formulation constituents in [8]. In the following, we briefly touch upon how we extract the relevant information and present it as a knowledge graph. We detail out the formulated products data we use to experiment. We then discuss the kinds of analyses necessary to generate new formulations in an aided manner.

We presented a human-in-the-loop way of generating desired formulations in [8], which we revisit to contrast with another way of generating formulations. This other way uses generic templates created for specific product types as the starting point. Although the core requirements from a system standpoint remain the same, the template-based approach enables a global view of the requisite constituents for a specific type of a formulated product. We present these and other key differences in these two ways of generating new formulations.

We begin by reviewing information extraction from our earlier work [8] next.

3.1 Information Extraction

As detailed in [8], we use regular expressions for extracting the ingredients, their weights, phases, and product name.

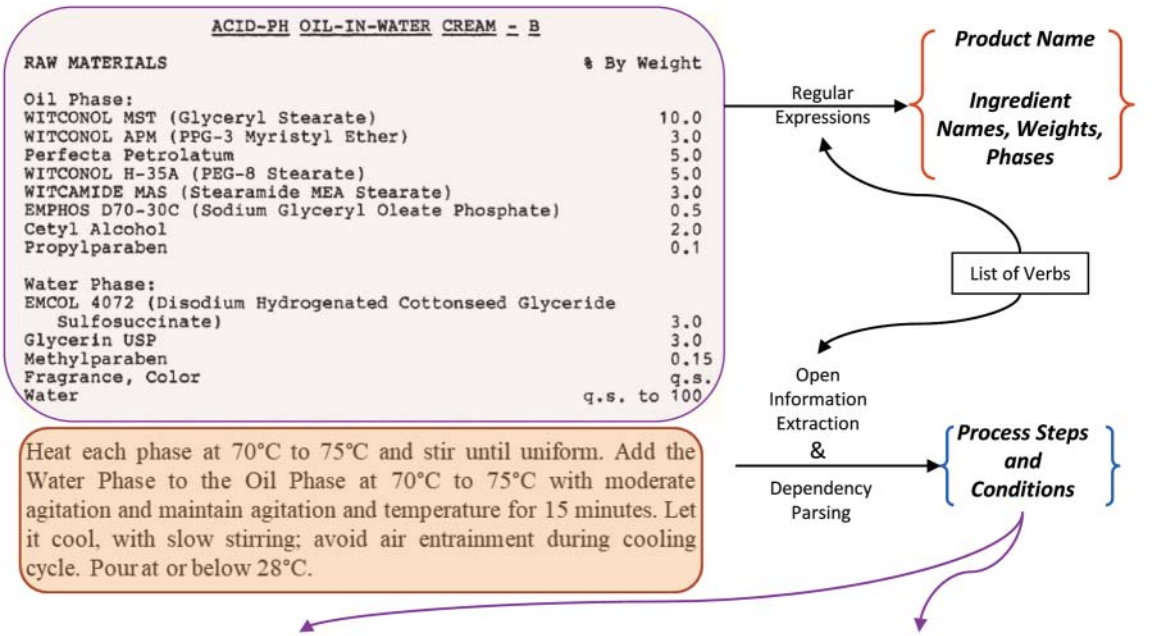
To extract recipe actions, we use an indicator list of verbs. We compile the list using the formulated products data detailed in Section 3.3. Verbs such as *maintain, heat, add, stir, moisturize, cool, extract, demineralize, mix, disperse, blend, emulsify, select, distill,* and *chelate* assume an essential role as process actions.

These verbs help separate the text representing the ingredients (the part containing the ingredient list has an absence of action verbs) and the text containing the procedure, which we call recipe text when used along with sentence boundary detection[9]. As illustrated in Figure 3, the ingredients occur in the part of the text that is NOT a set of sentences (whereas the recipe text is). The verbs also help in processing the recipe text to extract actions and conditions based on subject-verb-object structures of the sentences in the recipe text where the verb is one of the verbs from the list. We create Action-Mixture/Ingredient-Condition or (A-M-C) triples from every sentence. The collection of A-M-C triples from the recipe text is similar to the concept of the *action graph* described in several works such as [48, 49, 50, 51, 52].

We treat the *ellipsis* problem, references to ingredients and/or phases/ mixtures at earlier stages [51], using a stacking mechanism on top of information extractions techniques. Figure 3 shows an example formulation text along with the A-M-C structures obtained via open information extraction (Open IE[10]) and dependency parsing[11], respectively. For a detailed discussion and validation of the formulation constituents' extraction, the reader is requested to refer to [8].

We discuss the extraction of additional information such as synonyms and functionalities of ingredients from online sources in Section 3.3. Extracting these details does not require specialized information extraction techniques and relies mainly on packages for accessing Web pages[12] and pulling requisite data[13].

[9] Spacy Sentence Boundary Detection https://spacy.io/usage/spacy-101
[10] AllenNLP Open IE https://demo.allennlp.org/open-information-extraction
[11] Spacy Dependency Parser https://spacy.io/usage/linguistic-features/#dependency-parse
[12] Extensible Library for Opening URLs https://docs.python.org/3/library/urllib.request.html
[13] Scrapping Information from Web Pages https://pypi.org/project/beautifulsoup4/



Action	Mixture	Condition
Heat	Each phase	At 70°C to 75°C
Stir	Each phase	Until uniform
Add	water phase	To the oil phase at 70°C to 75°C
Maintain	Each phase + water phase + oil phase	Agitation and temperature for 15 minutes
Let	Each phase + water phase + oil phase	Cool
Avoid	Each phase + water phase + oil phase	Air entrainment during cooling cycle
Cooling	Each phase + water phase + oil phase	Cycle
Pour	Each phase + water phase + oil phase	At or below 28°C

Action	Mixture	Condition
Heat	Each phase	At 70°C to 75°C
Stir	Each phase	Until uniform
Add	Water phase + Oil phase	The water phase to the oil phase at 70°C to 75°C with moderate agitation
Maintain	Each phase + Water phase + Oil phase	Agitation and temperature for 15 minutes
Let	Each phase + Water phase + Oil phase	Cool with slow stirring
Avoid	Each phase + Water phase + Oil phase	Air entrainment
Pour	Each phase + Water phase + Oil phase	At or below 28°C

Figure 3. Extraction of formulation constituents.

We refer to the concepts in the domain model in Figure 4 in Section 3.4 when discussing the analyses, and we need to conduct to enable aided generation of formulations. We refer to the top-level type of formulated products such as cosmetic products or paints as the *FormulationType*. A specific category of such as creams or lotions within a *FormulationType* of cosmetic products is referred to as *FormulationCategory*. Within a *FormulationCategory* like creams, there could be numerous formulations such as face creams, anti-aging creams, and acne creams, as a *Formulation*. *Ingredient* instances partake in *Formulation* instances.

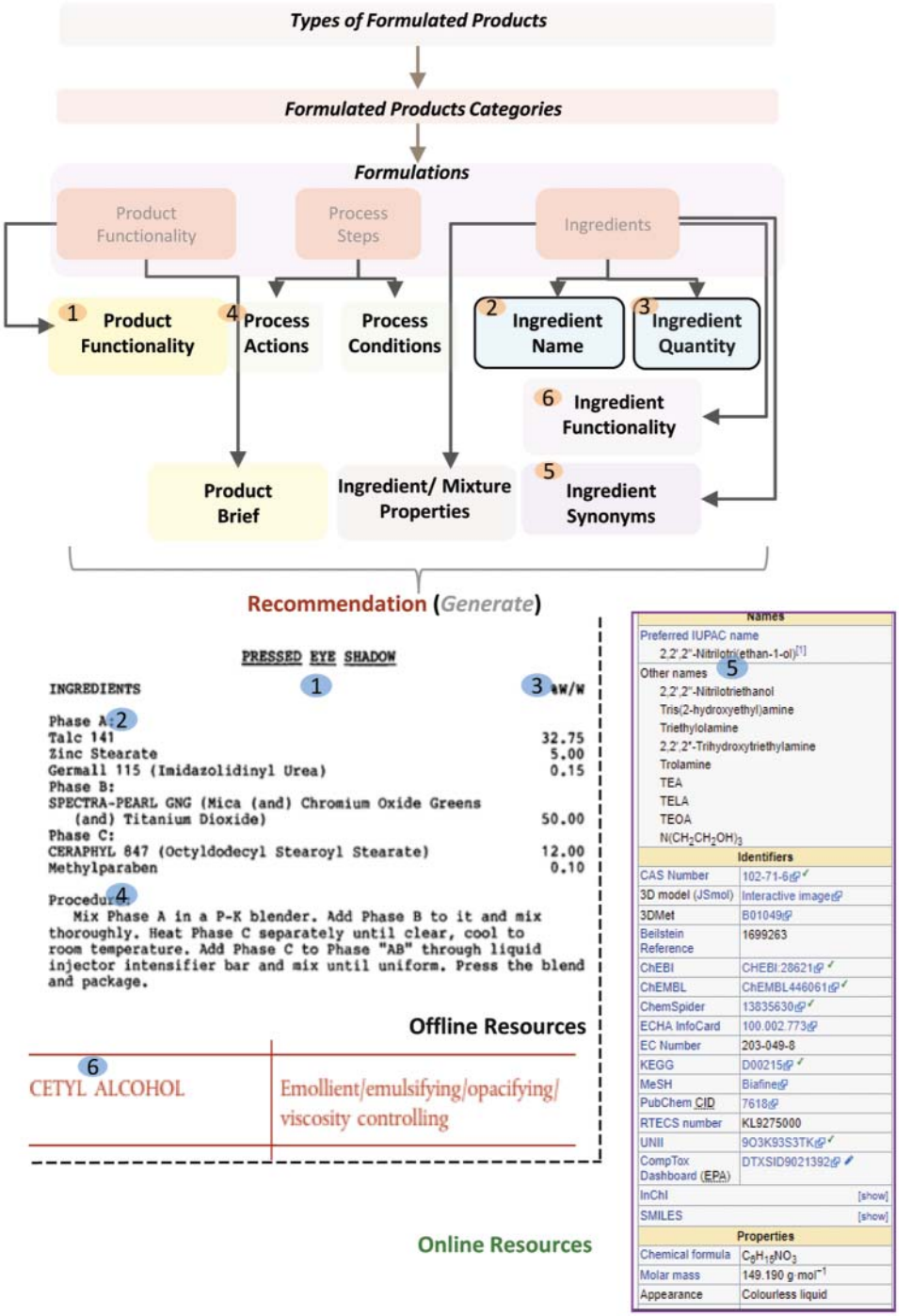


Figure 4. Graph domain model for formulated products data.

3.2 Graph Representation

Earlier in Figure 2, we showed the kinds of knowledge necessary for formulated products design. We also briefly touched upon the information extraction techniques we use to extract relevant data from offline and online sources in the previous section. Using the details presented in Figure 2, we can determine what we can refer to as a domain model or a schema for a graph database.

The bottom of Figure 4 shows formulation constituents and other details from offline and online sources of information about formulated products. The top of Figure 4 shows the schema underlying the graph database we use to store the extracted details, which we derive from Figure 2. Numbers 1–6 show how various extracted details map to specific concepts in the schema. Note that currently, we do not extract ingredient/mixture properties like values of solubility parameter, viscosity, and levels of spreading. Although with the proposed integration of *generate*, *make*, and *test* steps for formulated products, depending on the requirement, these details can be similarly obtained from online sources. Or these details can be computed and obtained from in-silico models as needed, as discussed later in Section 4.

We experiment with Neo4j[®] and RDF[®], which implement labeled property graphs and triple stores, respectively. For querying the Neo4j database, we use the Cypher[®] query language, and for triple store representation, we use SPARQL[®] (Figure 5). It is possible to implement the domain model or the schema shown at the top of Figure 4 in multiple ways in both Neo4j and RDF. For instance, in [8], we describe a Neo4j implementation in which we do not deduplicate the ingredients and use separate lists of ingredient synonyms and functionalities. In the RDF-based implementation, we include these details in the graph itself while also deduplicating ingredient details. The same ingredient appears in several formulations and is stored only once and referred. We show an example of such an implementation arrangement in Figure 6 in Section 3.4.

Figure 5 shows example queries in both Cypher and SPARQL on top of Neo4j and RDF databases, respectively. The first query returns action graphs or collections of A-M-C triples from *"all purpose"* cream instances in the database. The second and third queries return all formulations containing the ingredient Cetyl Alcohol and its weights, respectively. Our observation is that it is easier to form SPARQL queries compared to Cypher queries. In contrast, it is easier to represent complex schema in a labeled property graph database like Neo4j. Since it is possible to convert either kind of graph to the other [53], the choice between the two becomes qualitative [54, 55] and boils down to ease of implementation in a particular context.

[®] Neo4j Graph Database https://neo4j.com/
[®] RDF https://www.w3.org/RDF/
[®] Cypher Query Language https://neo4j.com/developer/cypher/
[®] SPARQL Query Language for RDF https://www.w3.org/TR/rdf-sparql-query/

Get action graph of all 'all purpose' creams	Get the formulations containing 'Cetyl Alcohol' as one of the ingredients	Get quantity of all ingredients of the name 'Cetyl Alcohol'
Cypher-Neo4j		
`MATCH (f:Formulation)-[:HasRecipeStringRepr]->(r:RecipeActionGraphStringRepr) WHERE f.name CONTAINS "all purpose" RETURN f.name, r.repr`	`MATCH (f:Formulation)-[:HasIngredient]->(ingd:Ingredient) WHERE ingd.name CONTAINS 'Cetyl Alcohol' RETURN COUNT(f.name) as nnumFormulations, collect(f.name) as Formulations`	`MATCH (f:Formulation)-[:HasIngredient]->(ingd:Ingredient) WHERE ingd.name CONTAINS 'Cetyl Alcohol' RETURN f.name as Formulation, ingd.name as IngredientName, ingd.quantity as WeightQT`
SPARQL-RDF		
`prefix rel: <http://www.w3.org/rel#> prefix node: <http://www.w3.org/node> SELECT ?recipeText where{ ?formulationID rel:hasFormulationName ?formulationName FILTER regex(?formulationName, "ALL PURPOSE CREAM", "i") ?formulationID rel:HasRecipeText ?recipeText }`	`prefix rel: <http://www.w3.org/rel#> prefix node: <http://www.w3.org/node> SELECT ?formulationName where{ ?formulationID rel:hasFormulationName ?formulationName . ?formulationID rel:hasIngredientWeight ?IngWeightID . ?IngWeightID rel:hasIngredient ?IngID . ?IngID rel:hasIngredientName ?ingName FILTER regex(?ingName, "cetyl alcohol", "i") }`	`prefix rel: <http://www.w3.org/rel#> prefix node: <http://www.w3.org/node> SELECT ?formulationName ?quantity where{ ?formulationID rel:hasFormulationName ?formulationName . ?formulationID rel:hasIngredientWeight ?IngWeightID . ?IngWeightID rel:hasIngredient ?IngID . ?IngWeightID rel:hasWeight ?quantity . ?IngID rel:hasIngredientName ?ingName FILTER regex(?ingName, "cetyl alcohol", "i") }`

Figure 5. Examples of queries based on domain model in Figure 4 in Cypher and SPARQL.

3.3 Formulated Products Data

We demonstrate our approach using 410 cream formulations obtained from Volumes 1 to 8 of the book *Cosmetic and Toiletry Formulations* by Flick, E. W. (1989–2014) [44] and 67 paint formulations obtained from another such book on paints by the same author [45].

A total of 2,633 ingredients exist in 410 cosmetic formulations, out of which 1,086 are unique, and 333 ingredients repeat more than once. In the case of paint formulations, we found 303 unique ingredients.

For ingredient synonyms, we used sites such as Wikipedia®, PubChem®[56], Chebi®, and ChemSpider®. We extracted the following details:

- IUPAC (International Union of Pure and Applied Chemistry) name, synonyms, Pub- Chem CID, and the uses section from Wikipedia.
- Chemical formula and PubChem CID from PubChem.
- We obtain the link to Chebi and ChemSpider from Wikipedia and collect synonyms from these sites.

We also extracted ingredients functionality, esp. for cosmetic products from other online sources®. On average, every ingredient was found to have 12 synonyms and at least 2 functionalities associated with it.

Having collected these data and stored them as a graph, we need to carry out several additional analyses before generating formulations. We discuss these analyses next.

® E.g., cetyl alcohol entry at Wikipedia https://en.wikipedia.org/wiki/Cetyl_alcohol
® at PubChem https://pubchem.ncbi.nlm.nih.gov/compound/1-Hexadecanol
® at Chebi https://www.ebi.ac.uk/chebi/searchId.do?chebiId=16125
® ChemSpider http://www.chemspider.com/Chemical-Structure.2581.html
® Cosmetics Ingredient Functionalities: https://bit.ly/3rCXnGz

3.4 Analyses for Formulated Product Variants

The primary method of coming up with a new formulation for a given formulated product type with specific properties consists of manually deciding and gathering the requisite ingredients based on their functionalities and applying specific actions at specific conditions.

We should be able to relate the functionalities of an ingredient established previously to any of its synonyms. This requirement is more of an implementation-level detail. We address it by storing the ingredient and its synonyms and functionalities (as shown for an example ingredient in Figure 6) and referring to a single ingredient node whenever its synonyms are part of a formulation.

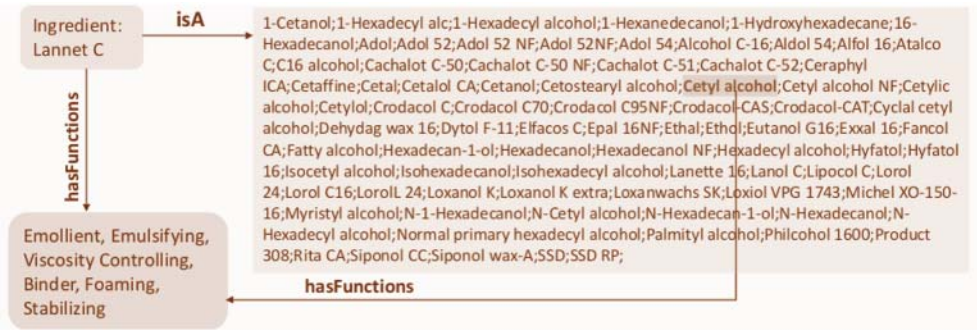


Figure 6. Relating ingredient synonyms and functionalities.

To enable the generation of formulations, we need to carry further analyses described next:

1). **Phase Naming Reconciliation** Formulations collected from several sources are likely to name the phases or mixtures in different ways. We need to reconcile the correct naming of phases to relate the ingredients and functionalities to correct phases. We assume phases to be *correct* when the phase names are *named* (as in water phase or oil phase in cosmetic formulations) in most formulations of a *FormulationCategory* available in the data, as against phases such as *phase* A and *phase* B. If the data contain no formulations with appropriately named phases or contain a formulation with no phases, we obtain the most common named phases by consulting with the expert before starting the reconciliation.

We infer the correct phases whenever they are missing, using the mentions of the same ingredients in other formulations where they are part of a specific phase (such as water phase or oil phase in case of creams or other cosmetic products). We show an example of phase naming reconciliation in Figure 7.

In Figure 7, vol* indicate different formulations. Having seen the ingredients *Propylene Glycol* and *Cetyl Alcohol* in water and oil phases respectively in other formulations of the same *FormulationCategory*, it is possible to ascribe them to specific phases when the formulation text does not explicitly refer to these phases or refers to them with arbitrary phase naming, such as *Part A*, as shown in Figure 7. It is also possible to find a particular ingredient such as a fragrance that is used in addition to other ingredients in named phases but not as part of these phases. We call such ingredients parts of additional phases.

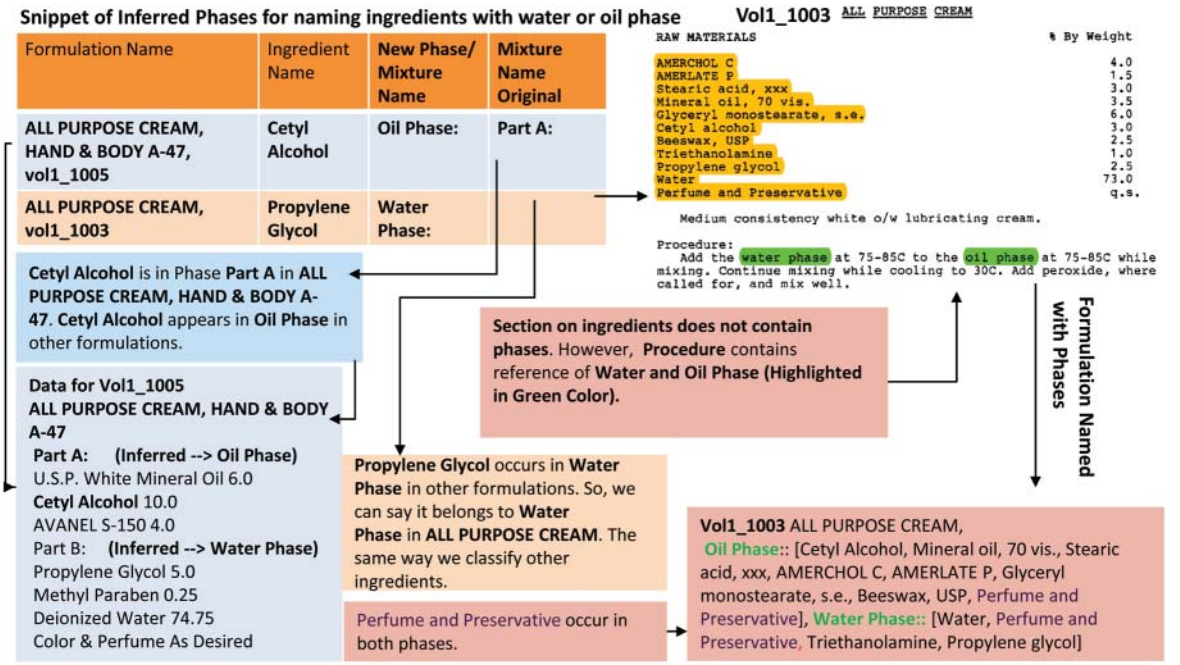


Figure 7. Phase naming reconciliation.

2). **Instruction Sequencing** In generating a new variant from scratch or using a template to kick-start this process, we need to order the sequence of instructions correctly. To establish an ordered sequence of instructions, we use the A-M-C triples obtained from the recipe texts of formulated products of a specific *FormulationCategory*. We compute the counts of Action, Mixture, and Condition instances occurring at specific positions and assign the highest rank to the maximum count at a particular position.

When constructing the ordered sequence, we choose the action at $position_i$ that has the highest rank. In the same way, we choose a mixture and condition for $position_i$. If the highest rank for a $position_i$ is for more than two actions, we break the tie by taking the highest rank of action-mixture pairs.

3). **Functionality Ranking in Phases** As indicated above, ingredients (representing specific Functionality) occur in specific phases. Once we reconcile the phases' names, it is possible to map specific functionalities to the ingredients in the named phases. Many ingredients represent multiple functionalities. To associate a specific functionality from many possible functionalities of an ingredient, we assume that in a particular kind of a formulated product, an ingredient is often used for its *primary* functionality. In contrast, its *secondary* functionalities are of no consequence or optional as far as that specific kind of formulated product is concerned. Therefore, we rank these functionalities of an ingredient based on the percentage of functionality occurrences in various phases.

We calculate functionality counts (via ingredients) and take the percentage of the functionalities' occurrence in various phases. This step returns the functionalities that map to mainly water, oil, and additional phases for cosmetic products. We then categorize the functionalities into primary and secondary functionalities per phase based on count percentage; primary functionalities are the ones that occur more than 80% of the time, while secondary functionalities occur with less than 80% of the formulations. We use the 80% threshold based on our observations from recipe variants of many formulated products of the same *FormulationCategory*.

Figure 8 shows an example of computing *primary* and *secondary* functionalities for face creams. We conduct this analysis mainly for the template-based approach, although it is often helpful to the expert to know which functionalities are primary in a specific *FomulatonCategory* even when generating the formulation from scratch.

Functionalities per Phase	Count
Oil Phase: emollient	15
Oil Phase : emulsifying	11
Water Phase: humectant	6
Oil Phase: preservative	5
Water Phase: preservative	5
Additional Phase: smoothing	5
Oil Phase: viscosity controlling	4
Water Phase: solvent	4
Water Phase: moisturizing	4
Additional Phase: film forming	4
Additional Phase: antiaging	4

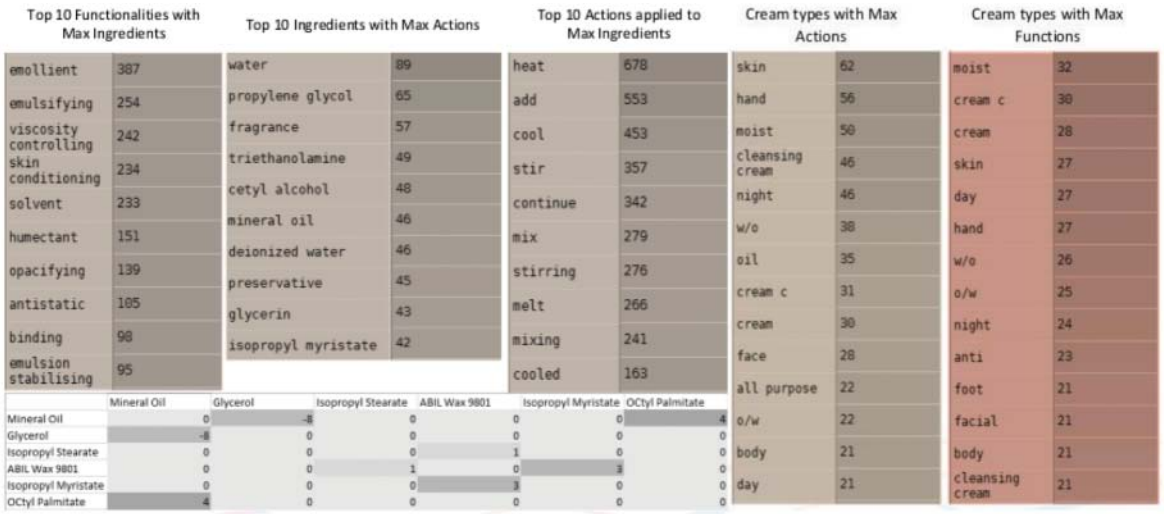
Primary Functionalities		
Phases	Example Ingredients	
Oil Phase	Miglyol : 0.3-20.0	Emulsifiers
	Anhydrous lanolin: 2.0-20.0	Emollient
	Phenonip: 0.3-qs	Preservative
Water Phase	Phenonip: 0.3-q.s.	Preservative
	Karion F liquid: 3.0-5.0	Humectant
Additional Phase	Collagen CLR: 5.0-10.0	Smoothing

Secondary Functionalities		
Phases	Example Ingredients	
Oil Phase	Cutina MD: 3.0-6.0	Viscosity Controlling
Water Phase	Water: 8.0-q.s.	Solvent
	Karion F liquid: 3.0-5.0	Moisturizing
Additional Phase	Collagen CLR: 5.0-10.0	Film-Forming
	Collagen CLR: 5.0-10.0	Anti-aging

Figure 8. Ranking functionalities of ingredients per phase; Example of face cream ingredients.

4). **Ingredient Correlations** To help the expert decide which representative ingredients to choose in combination either in a standalone manner or as members of named phases, we perform several correlation analyses.

Figure 9 shows the co-occurrences of ingredients or lack thereof, esp. in the same phase or mixture. Such clusters of ingredients are useful because the membership of an ingredient within a specific phase can inform the choice of other ingredients. Functionalities such as emollients and viscosity controlling dominate in cream formulations, whereas solvents happen to be prominent in paint formulations. *Water, Propylene Glycol, Fragrance, Triethanolamine* and *Cetyl Alcohol* are the most common ingredients in cream formulations, whereas *Water (deionized), Aqueous Ammonia,* and plasticizers like *Santicizer 160* are prominent in various paint formulations.



Top 10 Functionalities with Max Ingredients

emollient	387
emulsifying	254
viscosity controlling	242
skin conditioning	234
solvent	233
humectant	151
opacifying	139
antistatic	105
binding	98
emulsion stabilising	95

Top 10 Ingredients with Max Actions

water	89
propylene glycol	65
fragrance	57
triethanolamine	49
cetyl alcohol	48
mineral oil	46
deionized water	46
preservative	45
glycerin	43
isopropyl myristate	42

Top 10 Actions applied to Max Ingredients

heat	678
add	553
cool	453
stir	357
continue	342
mix	279
stirring	276
melt	266
mixing	241
cooled	163

Cream types with Max Actions

skin	62
hand	56
moist	50
cleansing cream	46
night	46
w/o	38
oil	35
cream c	31
cream	30
face	28
all purpose	22
o/w	22
body	21
day	21

Cream types with Max Functions

moist	32
cream c	30
cream	28
skin	27
day	27
hand	27
w/o	26
o/w	25
night	24
anti	23
foot	21
facial	21
body	21
cleansing cream	21

	Mineral Oil	Glycerol	Isopropyl Stearate	ABIL Wax 9801	Isopropyl Myristate	OCtyl Palmitate
Mineral Oil	0	-8	0	0	0	4
Glycerol	-8	0	0	0	0	0
Isopropyl Stearate	0	0	0	1	0	0
ABIL Wax 9801	0	0	1	0	3	0
Isopropyl Myristate	0	0	0	3	0	0
OCtyl Palmitate	4	0	0	0	0	0

Top Five Ingredients in Paints

Top Five Ingredients based on Type of the Paint

		Water based paint formulation		Enamels		Coatings and topcoats		Coil coatings		Top 5 Actions	
Water	84	Water	45	Water	19	Water	19	Foamaster TCX	3	add	586
Aqueous Ammonia	23	Aqueous Ammonia	11	Aqueous Ammonia	10	Minex 4	5	Acrysol 1-62	2	blend	170
Deionized Water	19	Ammonia	10	Deionized Water	8	Bentone LT	4	TI-Pure R-960	2	grind	167
Ammonia	14	Butyl Carbitol	8	sec-Butanol	6	Butyl Carbitol	4	Diethylene glycol monomethyl ether	2	Mix	75
Butyl Carbitol	14	Deionized Water	8	Intercar Cobalt	5	Santicizer 160	4	Rhoplex AC-658	2	mill	56

Figure 9. Correlations in ingredients, functionalities, and actions for formulations (at the top, for Creams [8]; at the bottom, for Paints).

Similarly, *Heat, Add,* and *Cool* are some of the most frequently occurring actions in cream formulations, while *Add, Blend, Mix,* and *Mill* are the most frequent actions in paint formulations.

We also relate functionalities of ingredients to specific kinds of formulations and thereby to formulation categories. For instance, massage cream instances tend to contain *antistatic*, *binding, buffering,* and *denaturant* functionalities prominently [8]. Similarly, chamomile cream instances tend to contain functionalities such as bulking, humectant, and plasticizer, among others.

5). **Weight Ranges of Ingredients** Weight ranges need to be associated with finalized ingredients, either when generating a new variant of a *FormulationCategory* from scratch or when using a template (since in generating a template, we need to associate weight ranges to the ingredients). We calculate the minimum and maximum values of quantities for each ingredient I_j and present as a weight range $w_{min} - w_{max}$.

With these analyses in place, generating new variants of a *FormulationCategory* either from scratch or starting with templates is relatively straightforward. In the next section, we show how we achieve these.

3.5 Generating Formulation Variants from Scratch

At this point, we essentially transform a predominantly manual set of steps into an aided set of steps as described below:

1). Given a specific kind of *FormulationCategory*, the formulation design system queries the functionalities it usually contains and shows them to the expert.
2). For each functionality, the system presents to the expert all the *Ingredient* instances associated with it and the weight ranges of these instances.
3). The expert finalizes the set of ingredients by consulting primary and secondary functionalities that this *FormulationCategory* generally contains and clusters of ingredients that occur together in the named phases of historical formulation of this *FormulationCategory*.
4). The system then queries the actions and conditions generally performed on each finalized ingredient in a standalone manner or as a part of a mixture from the A-M-C structures from the stored instances and applies instruction sequencing. The expert finalizes the sequence of instructions.

For a detailed walk-through of a face cream variant generation from scratch, we request the reader to refer to [8].

3.6 Product Template Generation

Generating a template for a specific *FormulationCategory* is now easily achievable, given that the formulation data already contain few instances of that *FormulationCategory*. Assuming that all the previously listed analyses have been performed for the constituents of a specific *FormulationCategory*, we generate a template for it as follows:

1). The templating system queries the primary and secondary functionalities for that *FormulationCategory*.
2). The system then queries the ingredients representative of the various functionalities and associates the ingredients' weight ranges to each functionality. If the weight ranges differ drastically for the ingredients representative of the functionality, the system adds a marker indicating this situation.
3). The system arrives at an ordered sequence of instructions using the A-M-C structures of the formulations as described earlier.

Figure 10 shows an example of the existing template[23] and the template generated by our templating system for face creams based on six face cream formulations in our data.

To aid the expert in instantiating the template, we display related information to the expert, as shown in Figure 11. This information shows possible candidates for substitution over functionalities in the template, along with other ingredients known to co-occur and the description of the functionalities.

[23] Available at MakingCosmetics https://bit.ly/3aPrODt

Face Cream Template

Must Use Ingredients	Typical Use Level	Example Ingredients
Distilled Water	55%-75%	Water Mixed with Botanical Extracts
Emulsifiers	2%-6%	CreamMaker Blend, Polysorbates
Emollients	10%-35%	Plant Oils, Natural Butters
Thickeners	0.1%-2%	Natural Waxes, Stearyl Palmitate
Optional Ingredients		
Active Ingredients	1%-10%	Anti-Aging Ingredients, Skin-Whitening Ingredients
Fragrance	0.1%-1.0%	
Preservatives	0.1%-1.0%	Benzylalcohol-DHA

Method:
Water phase: Thickener is dispersed in cold or warm water at 75°C-80°C (depending on type of thickener) under stirring until a homogeneous gel is formed. Combine then with separately mixed oil phase that contains emulsifiers and emollients which have also been melted and heated to the same temperature. Mix under intensive stirring until emulsion is formed. Then mix gently while emulsion is being cooled. Sensitive components like active ingredients, fragrances and preservatives are added after the mixture has been cooled (40°C-30°C) to keep their properties intact.

Generated Template for Face Cream

Primary Functionalities	
Phases	Functionalities
Oil Phase	Emulsifier: 0.3-20.0
	Emollient: 2.0-20.0
	Preservative: 0.3-qs
Water Phase	Preservative: 0.3-q.s.
	Humectant: 3.0-5.0
Additional Phase	Smoothing: 5.0-10.0

Secondary Functionalities	
Phases	Functionalities
Oil Phase	Viscosity Controlling: 3.0-6.0
Water Phase	Moisturizing: 3.0-5.0
Additional Phase	Film Forming: 5.0-10.0
	Anti-aging: 5.0-10.0

Recipe:
1. Warm water phase to about 70°C.
2. Bring oil phase to about 70°C.
3. Combine and stir water phase and oil phase.
4. Cool to about 35°C.
5. Add additional phase.

Figure 10. A known *Face Cream* template (on the left) and the generated template (on the right).

List of Ingredients with Specific Functionalities and Description

Functionality	List of Ingredients	Description of Functionality
EMULSIFYING	• Miglyol • EMEREST 2400 Glyceryl Stearate • EMERWAX 1266 Emulsifying Wax BP Type	Promotes the formation of intimate mixtures of non-miscible liquids by altering the interfacial tension.
EMOLLIENT	• LANACET 1705 Acetylated Lanolin • Anhydrous lanolin	Softens and smooths the skin.
HUMECTANT	• Propylene Glycol • Karlon F liquid"	Holds and retains moisture.
PRESERVATIVE	• Propylparaben • Methylparaben • Phenonip	Inhibits primarily the development of micro-organisms in cosmetics. All preservatives listed are substances on the positive list of preservatives (Annex VI to the cosmetics Directive).
SMOOTHING	• Collagen CLR • Elastin CLR	Seeks to achieve an even skin surface by decreasing roughness or irregularities.

Correlations from the Same Phases

Functionality	Ingredient 1	Ingredient 2	Count
Emollient	Anhydrous lanolin	Cetiol V	2
Emulsifying	Eumulgin B	Miglyol 812	2
Preservative	EMERESSENCE II60 ROSE ETHER Phenoxyethanol	Methylparaben	2

Correlations from Different Phases

Functionality	Ingredient 1	Ingredient 2	Count
Emulsifying	EMEREST 271 7 PEG-I 00 Stearate	EMERWAX 1266 Emulsifying Wax BP Type	1
Humectant	EMEREST 2380 Propylene Glycol Stearate	Propylene glycol	1

Figure 11. Aid to experts in instantiating product template.

3.7 Current Limitations and Future Work in Aided "Generate" Step

Our current implementation of the aided *generate* step has the following limitations and ways forward:

- As discussed in [8], the information extraction techniques we use tend to fail in specific linguistic cases of recipe text, like gerund verb forms (mixing in "disperse . . . using high-speed mixing") and passive instructional sentences. Although such cases are rare, this problem leads to incomplete A-M-C structures. Since formulated product recipes are similar to cooking recipes, we are working on a transfer learning approach. We plan to train an Open IE model on a large cooking recipe data set [57, 58] and then fine-tune it to our data.
- When computing weight ranges, we consider the weights given for an ingredient in various formulations. Since the weights are often in some proportion in a particular recipe, we need to keep track of these proportions (such as in a specific type of cream, the emulsifier is *x* parts, an emollient is *y* parts, and the thickener is *z* parts). We plan to present these proportions as additional information to the expert. This forms part of our ongoing work.
- We categorize functionalities into primary and secondary functionalities based on occurrence percentage, which is influenced by historical recipes in our data set and may be incorrect. It is possible to correct an incorrect categorization by consulting with an expert in a one-time check.

3.8 Addressing Concerns in Formulation Generation

3.8.1 Human-machine Coordination and Assistance of Experts

We implement the human-in-the-loop formulation generation considering both i) the expert's inputs and ii) the assistance provided to the expert. We can consider two main stages of expert interaction: building the knowledge graph (before using the knowledge graph in formulation generation) and the formulation generation process. Before the population of the knowledge graph, we consult with the expert at various stages, such as phase naming reconciliation (Section 3.4), asking the expert the known proportions of ingredients as opposed to weights (Section 3.7), and checking functionality categorization with the expert (Section 3.7). These are the steps where the expert is supposed to provide inputs which reflect in the knowledge graph.

During the formulation generation process (Section 3.5), the system assists the expert at various stages. The expert using the system's information and his/her knowledge and experience chooses from the suggested ingredients, weights, and process steps. Thus, our system both inquires the expert to enrich the knowledge graph and interacts with the expert during the formulation generation process.

3.8.2 Controlling Weights and Proportions of Ingredients

Although ingredient weights and proportions are core secrets for the formulated product companies, we propose using *offline* sources such as historical formulations to collect the information on weights and proportions of ingredients. We describe the details in Section 3.1 and Figures 3 and 4. Figure 4 distinguishes

the various constituents obtained from online and offline sources and it shows that we obtain the ingredient weights/proportions from offline sources. Online sources generally do not contain this information as they describe information specific to an ingredient irrespective of the type of formulated product in which it can be used.

Another critical concern is *deciphering the possible effect of one ingredient over the other ingredients' functionalities*. We approach this concern in two ways. First, we use historical formulations from published sources as in Section 3.4 to compute various correlations. Second, later in Sections 4.2 and 4.3, we propose to use in-silico testing, wherein physics-based modelling (typically at molecular scale) can be used to investigate the co-relation of different ingredients by performing simulations with varying ingredient proportions and assessing their effect on desired properties. It is possible to add this information to the knowledge graph as additional information for an ingredient to be available for the expert's consideration during formulation generation.

3.8.3 Choice between Formulation Generation Methods

We offer the two methods for formulation generation (described in Sections 3.5 and 3.6, respectively) to the expert who can use either one or both methods simultaneously. The template-based approach gives the expert a comprehensive view of the functionalities needed for the formulation. The method to generate formulation from scratch offers suggestions at every step of generation. Suppose the expert is not aware or has not worked on the formulated product type under consideration. In that case, we expect the expert to refer to the template as a *ready reckoner*. If the expert has experience dealing with the formulated product type under consideration, he/she may choose to use the step-by-step formulation generation. Since we do not enforce one method's use over the other and let the expert choose, we believe that a decision-making mechanism is not required to decide which method to use. The expert can utilize both methods for their intended purpose.

4. INTEGRATING *GENERATE, MAKE, TEST* STEPS IN FORMULATION DESIGN

The realization of any formulated product usually takes place by following a 3-step process viz. (a) *generate* (b) *make* and (c) *test*. The *generate* step comes up with the base recipe for a given product on the desired properties intended for the product, as shown in the previous section.

In a typical new product development cycle, the base recipe as constructed at the *generate* step is merely a starting point. It needs to be refined further on various aspects (e.g., what is the lowest proportion in which the costliest ingredient in the formulation can be used still maintaining the same effect). The *make* step involves making the formulation in the lab with a range of ingredient proportions and conditions, followed by conducting requisite testing for each formulation to assess various desired and required properties.

The design and test step results are scrutinized to identify the most optimum formulation (possibly 1/100's or 1/1000's), which would satisfy the cost-performance trade-off. This optimum formulation is the one that is finally considered for scale-up and becomes the final product.

Traditionally, all three steps have been performed manually in an experience-guided, document-centric, and experiment-heavy manner resulting in stretched cost and low efficacy and agility. With increasing demand and consumer awareness, reducing turnaround times, and stricter regulations, the formulated product industry needs to embrace digitalization in all three steps to derive real value.

With the base recipe for a cosmetic/coating formulation obtained from the *generate* step as the starting point, the following sections attempt to exemplify how digital intervention at the *make* and *test* steps would transform the traditional trial and error approach of formulated product design to a knowledge guided approach.

4.1 Automated "Make" of a Formulated Product Variant

The automated make step requires creating a detailed design of the experiment chart. Such a chart consists of various combinations of proportions of ingredients and conditions associated with actions; each combination corresponding to a unique formulation, which needs to be synthesized by the robotic equipment and tested subsequently.

There are five ingredients in the current case of a variant of face cream (five factors), as shown in Figure 13. We obtained this recipe by carrying out the generate step as detailed in [8]. Considering only the minimum and maximum values of weights for each ingredient (leaving out the action conditions for the simplicity of explanation), there are only two levels, which amounts to a minimum of 2^5 experiments to be carried out.

To carry out a more detailed exploration, weight ranges for each ingredient could be divided into several intervals (equal or unequal); for example, 10%~35% for isopropyl myristate could be divided into five intervals difference of 5%. For instance, even if we consider two intermediate levels for each ingredient (excluding mix-max, hence total four levels), the complete factorial design of experiments (excluding conditions for actions) for the face cream formulation would amount to 4^5 = 1,024 experiments.

It is important to note here that, to truly reject or accept a particular combination of weight proportion and action condition, it would be required to subject the corresponding formulation to testing; thus, 1,024 formulation *make* experiments also translate to an equivalent number of testing evaluations.

These numbers provide enough motivation for the industry to adopt high throughput formulation-making procedures (automated laboratories) (automated make) and, more so, exploit the concept of autonomous laboratories in recent times. These numbers represent the worst-case scenario in which no heuristic, experience, or prior knowledge is available regarding ingredients, their effects, and roles in the final product's properties.

On the other hand, if we were to follow the autonomous route, which, as explained earlier, poses the problem of finding the best combination of ingredient proportion and action conditions as an experimental design problem, we can expect to reach the most optimum formulation in comparatively a smaller number of trials. Many methods can solve this problem of finding the global optimum of a non-convex objective function viz. random search [59, 60], systematic grid search [61], and recently Bayesian optimization-based techniques [62, 63, 64].

In the face cream case, consider that the objective function to be minimized is the difference between the desired property (as derived from the product brief) and the obtained property (as calculated from the corresponding test procedure for the property (e.g., viscosity, density, and pH). Given a starting value for each decision variable (which can be any value including or between the range (upper-lower bounds), the random search algorithm suggests the next set of decision variables. These can be used to prepare the formulation according to the recipe, followed by conducting the requisite tests. The above procedure is repeated till the error between the desired and obtained property is minimized. The same discussion applies to the external coating variant shown in Figure 14.

4.2 Completing Integrated Design of a Face Cream Variant with In-silico Testing

Human skin is a layered organ (epidermis, dermis, hypodermis, etc.). The topmost layer of the epidermis viz stratum corneum (SC) acts as a barrier and exhibits selective permeation behavior. Thus, cosmetic and toiletry formulations (creams, soaps, etc.), especially those used for topical applications, often find a need to contain supporting ingredients (e.g., permeation enhancers) that assist the active ingredient(s) in breaching the skin barrier. Hence, to design effective enhancers, it becomes necessary to understand various aspects of these chemicals' interaction with the SC.

At Tata Consultancy Services (TCS) Research, we have developed an in-silico model of human skin based on a multi-scale modelling framework [42]. This framework can simulate the interaction of molecules of interest with human skin, thereby providing insights to various crucial mechanisms at the molecular level and predicting properties of interest to aid in screening/designing these molecules.

For instance, in the face cream variant, shown in Figure 13, isopropyl myristate is the second most required ingredient and is supposed to function as an emollient. However, studies show that it also depicts permeation enhancer properties [65, 66].

Recently, Gupta et al. [43] have performed detailed coarse-grained (CG) molecular dynamics (MD) simulations to calculate partition and diffusion coefficients of permeation enhancers in the SC lipids. The study involved enhancers from different families viz fatty acids, esters, and alcohol at different concentrations 1%, 3%, and 5% w/v.

Figure 12 summarizes their study's findings in terms of key observations. They also provide the enhancement ratio and partition coefficient (log P) obtained from simulations as property values used as selection criteria for enhancers [43]. They also mention other observations that give more specific details about the enhancers'

interaction mechanisms with the SC lipid constituents. Figure 12 summarizes their observations for isopropyl palmitate (ISP).

Obsv.1	ISP has an enhancement ratio (ER) of 27.11 and a partitioning coefficient (log P) of 10.94 as obtained from CG MD simulations.
Obsv.2	ISP partitions completely from solvent to the SC lipid, is dispersed in the lipid layer and crosses the lipid bilayer.
Obsv.3	ISP does not create disordering in the SC lipid, infact with increasing concentration of ISP there is a reduction in the order parameter of SC lipid.
Obsv.4	Diffusion coefficient of ISP increases with concentration.
Obsv.5	ISP interacts significantly with free fatty acids followed by cholesterol and ceramides.

Figure 12. Observations for isopropyl palmitate (ISP).

Apart from such specific observations, they also state a very generic and key observation that "*small hydrophobic molecules partition well into the skin lipid layer and do not agglomerate. On the other hand, bigger hydrophobic molecules partition well and disturb the lipid layer packing significantly, but they sometimes form small clusters and limit permeation by diffusion rate.*".

Since cosmetic products are majorly used for beautification (anti-aging, anti-wrinkle, etc.), they contain active ingredients that interact with skin's mechanical properties (viscoelasticity, young's modulus, etc.). To understand the effects of ingredients on skins' mechanical behavior, Jayabal et al. [40] have recently developed a 1D viscoelastic model applied to experimental data for obtaining viscosity and modulus of elasticity of the skin. The authors also demonstrate that the same model can be extended to predict skin behavior when applied with a polymer layer on top. Polymers are often used in cosmetic formulations as thickening agents, emulsifying agents, creating protective films or barriers, and so on.

Figure 13 shows that property values (enhancement ratio, log P, diffusion coefficient, young's modulus, viscosity, etc.) obtained from these simulations should be added as additional information about ingredients in the knowledge graph. In contrast, the generic and ingredient specific qualitative observations could be used as heuristics which can be presented to the formulation chemist at the time of ingredient selection or template filling, thus genuinely achieving the knowledge-guided design of formulated products.

4.3 Completing Integrated Design of an External Coating Variant with In-silico Testing

External coatings (applied on automotive, airplanes, bridges, houses, machinery, etc.), although primarily used for decorative purposes (enhancing appearance by imparting color and/or gloss to a surface), are also designed to act as a protective cover for the surface beneath [67].

There are three major environmental factors viz heat, moisture, and radiation apart from rain, snow, microbial attack, mechanical stress, etc., against which external coatings are designed to provide protection. External coatings are usually applied in various stages across multiple layers and in a particular order, with each layer differing in composition, thickness, and purpose [68].

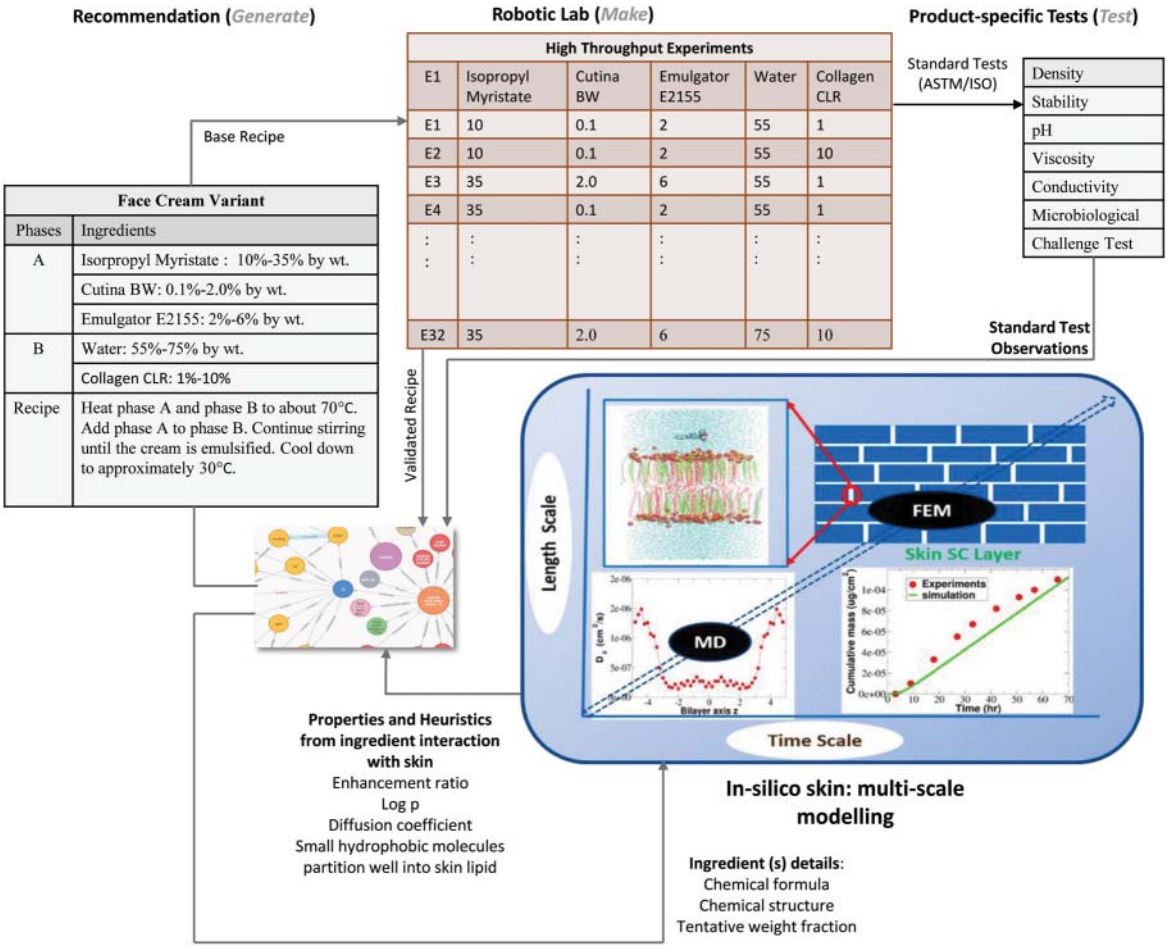


Figure 13. Integrated *Generate*, *Make*, and *Test* for a Face Cream variant. Note: In-silico skin: multi-scale modelling - reprinted (adapted) with permission from [42] Copyright (2017) American Chemical Society.

These coating systems are intended to retain their functionalities for the entire service life of the object on which they are applied. Thus, it is a norm to evaluate these formulations against the environmental factors by conducting elaborate weathering tests to assess their long-term performance.

Natural weathering tests are carried out in specific locations where formulations are exposed to extreme environmental conditions and are monitored for as long as five years[8]. However, current product development lifecycles are too short to allow waiting times of the order of years to get results from an experiment. Hence, accelerated weathering tests are more sought after; wherein one can induce environmental factors at a controlled rate and study the similar phenomenon in a time frame of days compared to years [69].

[8] Evaluations of Coatings https://bit.ly/3ryZCKJ

However, scientists believe that accelerated tests may not capture the precise degradation mechanisms during natural weathering. Environmental factors tend to interact with the external coating systems (especially with the binder and pigments) and degrade them over time, thus deteriorating performance.

One possible way to address the above challenges, that of prolonged testing (natural weathering) and unreliable results (accelerated weathering), is to model the degradation phenomenon by first-principle simulations or statistical methods (MD, density functional theory (DFT), Monte Carlo (MC), kinetic Monte Carlo (KMC), etc.). Makki et al. demonstrated the capability of a multi-scale simulation approach to imitate the photo-degradation process for a model polyester urethane coating system [70].

To account for the fact that the spectrum of time scales is substantial, across which the degradation takes place (chemical reactions—picoseconds and mechanical failure—years), they propose to couple an event-driven method (KMC) with a time-driven method (Dissipative Particle Dynamics (DPD) to accurately capture the physics.

These simulations allow calculations of important thermo-mechanical properties., e.g., glass transition temperature, storage modulus, loss modulus, thermal expansion coefficient, and crosslinking density at different degradation times.

Hinderliter et al. developed an MC-based methodology to model erosion of coating surface due to photon flux and utilized surface topography and chemistry changes to predict macroscopic properties like gloss, fracture toughness, and wetting angle [71]. An improved proximity-based molecular dynamics technique has been demonstrated for modelling crosslinking of thermoset polymers [72]. The authors used their methodology to calculate important material properties like glass transition temperature, stiffness, strength (stress *vs*. strain curves), and Poisson's ratio and capture the effects of curing temperature and crosslinking degrees on these properties.

Crosslinking imparts structural integrity and helps create a barrier for transporting foreign species (chemicals, moisture, etc.). Unreacted moieties in crosslinked coating resins lead to inhomogeneities and phase separations in the crosslinked coating film. At TCS Research, we have developed a multi-scale modelling-based approach to predict hyperelastic properties of elastomers at bulk level [73]. The approach involves obtaining microscopic properties by carrying out MD simulations of the crosslinked system and providing them as inputs to constitutive models for predicting the stress-strain response of the elastomer under different loading conditions.

Figure 14 shows that the properties and heuristics from ingredient interaction with the environmental factors can be added as additional information about ingredients in the knowledge graph and used as described above. For instance, the properties enlisted in the figure would help one understand different aspects of coating's performance, e.g., with knowledge of glass transition temperature, one can comment on water uptake, barrier properties, etc. At the same time, the variation in storage and loss modulus allows understanding the viscoelastic behavior. With the presence of in-silico polymer, acquiring such knowledge becomes inexpensive as compared to conducting lengthy experiments. It helps the formulation chemists to make informed decisions at *generate* step, thus accelerating product development.

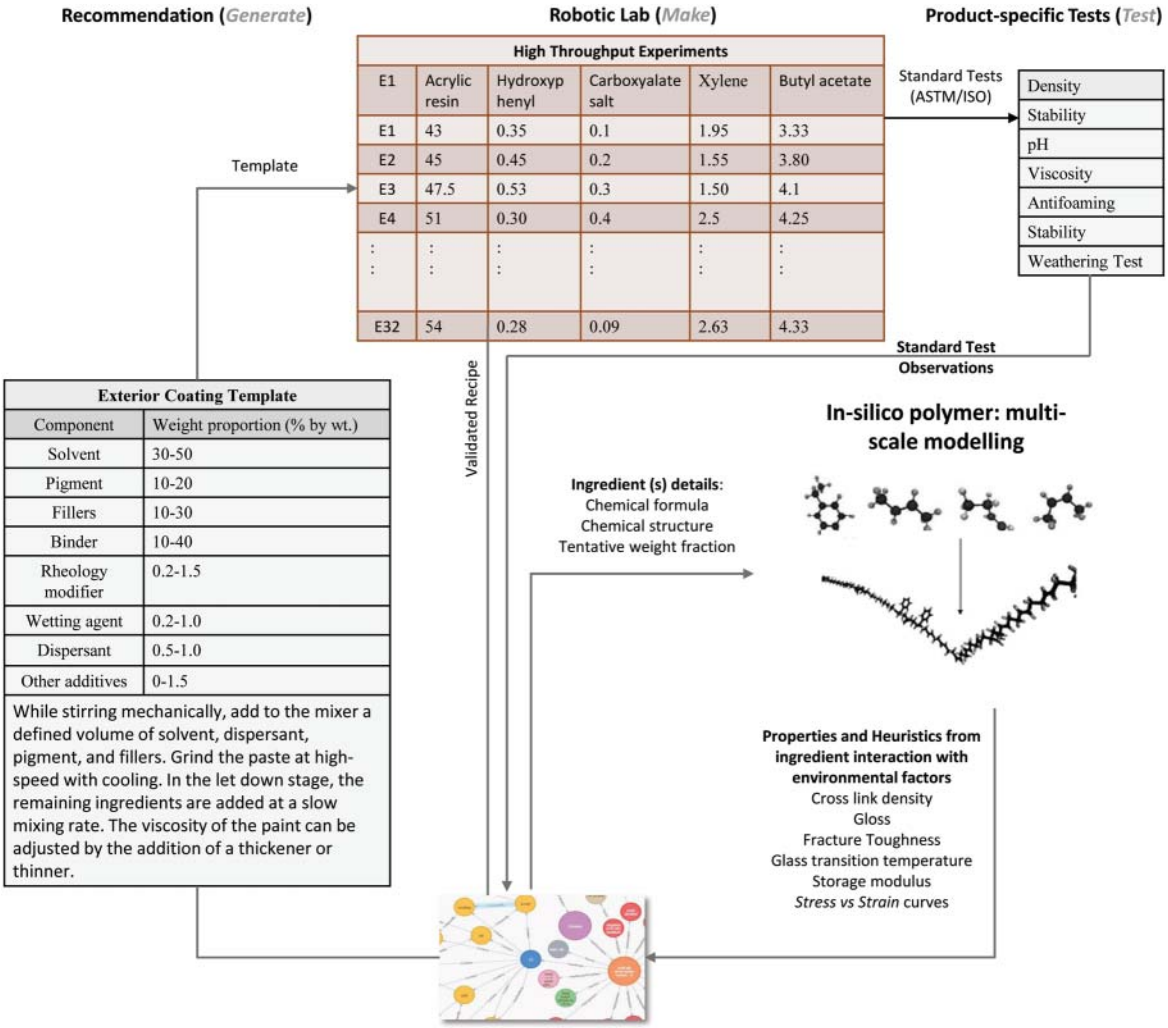


Figure 14. Integrated generate, make, and test for an External Coating variant. Note: In-silico polymer: multi-scale modeling - adapted and redrawn from [73] Copyright (2019) Taylor and Francis.

4.4 Summarizing Integrated Formulation Design with Knowledge Graph

The current state of practice involves the manual generation of recipes, manually operated making of candidate formulations, and third-party testing (as is the prevalent practice in many formulated product companies). In contrast, we proposed two instantiations of the aided *generate*, robotic *make*, and in-silico *test* steps, integrated by treating the knowledge graph as the central artifact. The *generate* step aids the expert in creating formulation variants while considering *test* step observations. The robotic *make* step stores the validated recipe in the knowledge graph. The knowledge graph continues to grow with these additional pieces of information, which often go unaccounted for in the current state of practice.

With this arrangement, we believe that the three steps in formulation design remain mindful of other steps and observations made in them. As mentioned earlier in Section 1, we assume interfacing systems between the three steps and the knowledge graph. Depending on the realizations of automated *make* and in-silico *test*, the implementation details may vary for such interfacing. However, with the aided, automated, and in-silico versions, respectively, the *generate*, *make*, and *test* steps can individually refer to the observations as knowledge in the next round of experiments.

4.5 Addressing Concerns in Integrated Formulation Design

4.5.1 Expert Contribution in Formulation Optimization

As discussed in Sections 4.2 and 4.3, the in-silico tests described involve performing physics-based simulations by modelling the system according to the intended outcome. Computational chemists, who are experts in modelling and simulation, need to decide on various factors such as the appropriate technique (molecular dynamics, Monte Carlo, density functional theory) and corresponding simulation parameters (time step, potential). Additionally, the experts need to interpret the simulation results by postprocessing the output from these physics-based models targeted at formulation optimization.

4.5.2 Verification of Information in Knowledge Graph

We are aware that various information pieces added to the knowledge graph (described in Section 3 and Sections 4.2 and 4.3) need to be verified. Currently, we rely on the correctness of information in published handbooks (offline) and the correctness of information about ingredients in specialty sites (online). In ongoing and future work, we plan to work on knowledge graph fact-checking and verification methods.

4.5.3 Validation of Proposed Experiments

We believe that the current manual steps, when aided with the knowledge graph and described in Section 3 and this section, will lead to cost reduction. Since currently, we have only a proposal for integrating the *generate*, *make*, and *test* steps, we have no experimental validation of cost reduction. This is part of our ongoing and future work.

5. CONCLUSION

Historical formulations and related data spread over offline and online resources present an opportunity to aid the expert in generating new formulations. We presented an approach to create and analyze a knowledge graph to provide recommendations for building a formulation recipe from scratch and using the template created to arrive at a viable formulation variant.

The knowledge graph we built can also be used as a connecting artifact between the *generate*, *make*, and *test* steps in formulation design. We show two instantiations of such an arrangement. In the first instantiation, the expert *generates* a face cream variant using the recommender system on top of the knowledge graph. The *make* step obtains the recipe and makes the formulation as described. The in-silico *test* step applies the digital skin model to test the requisite properties and feeds the observations back to the knowledge graph where the expert can refer to them as *heuristics* the next time. In the second instantiation, the expert starts with a template and follows the rest of the steps. In both examples, automated labs and in-silico models replace the traditional manually operated *make* and *test* steps. Our approach enables aided *generate* step with automated/robotic *make* and in-silico *test* in formulation design. We believe that such examples pave the way to an aided formulation design, where observations of each step can supplement and aid the other steps, esp. in iterated design steps.

AUTHOR CONTRIBUTIONS

S. Sunkle (sagar.sunkle@tcs.com), D. Jain (deepak.jain3@tcs.com), B. Rai (beena.rai@tcs.com), and V. Kulkarni (vinay.vkulkarni@tcs.com) conceptualized and ideated the integrated formulation design. S. Sunkle and D. Jain wrote and reviewed the whole paper. K. Saxena (krati.saxena@tcs.com), A. Patil (ab.patil2@tcs.com), T. Singh (singh.tushita@tcs.com), and S. Sunkle developed the knowledge graph, the recommender, and template generation systems.

DATA AVAILABILITY STATEMENT

The data sets generated during and/or analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.

REFERENCES

[1] EU formulation network deliverables 3.3 and 3.4, common vision and roadmap for formulated products (2016). Available at: https://bit.ly/3rBs7Yg. Accessed 20 April 2021

[2] Harper, P.M., Gani, R.: A multi-step and multi-level approach for computer aided molecular design. Computers & Chemical Engineering 24(2–7), 677–683 (2000)

[3] Conte, E., Gani, R., Ng, K.M.: Design of formulated products: A systematic methodology. AIChE Journal 57(9), 2431–2449 (2011)

[4] Gani, R., Ng, K.M.: Product design–molecules, devices, functional products, and formulated products. Computers & Chemical Engineering 81, 70–79 (2015)

[5] Zhang, L., et al.: An integrated framework for designing formulated products. Computers & Chemical Engineering 107, 61–76 (2017)

[6] Zhang, L., et al.: Advances in chemical product design. Reviews in Chemical Engineering 34(3), 319–340 (2018)

[7] Zhang, L., et al.: Chemical product design–recent advances and perspectives. Current Opinion in Chemical Engineering 27, 22–34 (2020)

[8] Sunkle, S., et al.: Information extraction and graph representation for the design of formulated products. In: Dustdar, S., et al. (eds.) CAiSE 2020, pp. 433–448. Springer, Berlin (2020)
[9] Hill, M.: Chemical product engineering—the third paradigm. Computers & Chemical Engineering 33(5), 947–953 (2009)
[10] Cussler, E.L., Moggridge, G.D.: Chemical product design. Cambridge University Press, Cambridge (2011)
[11] Martín, M., Martínez, A.: A methodology for simultaneous process and product design in the formulated consumer products industry: The case study of the detergent business. Chemical Engineering Research and Design 91(5), 795–809 (2013)
[12] Lee, C., Choy, K.L., Chan, Y.: A knowledge-based ingredient formulation system for chemical product development in the personal care industry. Computers & Chemical Engineering 65, 40–53 (2014)
[13] Sunkle, S., et al.: Generate and test for formulated product variants with information extraction and an in-silico model. In: Advanced Digital Architectures for Model-Driven Adaptive Enterprises, pp. 223–250. IGI Global, Hershey (2020)
[14] Chatterjee, P., Alvi, M.M.: Excipients and active pharmaceutical ingredients. In: Pediatric Formulations, pp. 347–361. Springer, Berlin (2014)
[15] Arrieta-Escobar, J.A., et al.: Incorporation of heuristic knowledge in the optimal design of formulated products: Application to a cosmetic emulsion. Computers & Chemical Engineering 122, 265–274 (2019)
[16] Lee, C.K.H.: A knowledge-based product development system in the chemical industry. Journal of Intelligent Manufacturing 30(3), 1371–1386 (2019)
[17] Taifouris, M., et al.: Challenges in the design of formulated products: Multi-scale process and product design. Current Opinion in Chemical Engineering 27, 1–9 (2020)
[18] Bernardo, F.P., Saraiva, P.M.: A conceptual model for chemical product design. AIChE Journal 61(3), 802–815 (2015)
[19] Picchioni, F., Broekhuis, A.: Material properties and processing in chemical product design. Current Opinion in Chemical Engineering 1(4), 459–464 (2012)
[20] Arrieta-Escobar, J.A., et al.: Integration of consumer preferences and heuristic knowledge in the design of formulated products: Application to a cosmetic emulsion. Computer Aided Chemical Engineering 46, 433–438 (2019)
[21] Conte, E., et al.: Design of formulated products: Experimental component. AIChE Journal 58(1), 173–189 (2012)
[22] Lindsey, J.S.: A retrospective on the automation of laboratory synthetic chemistry. Chemometrics and Intelligent Laboratory Systems 17(1), 15-45 (1992)
[23] Porte, C., et al.: Automation and optimization by simplex methods of 6-chlorohexanol synthesis. Process Control and Quality 4(8), 111–122 (1996)
[24] Wagner, R.W., et al.: Investigation of cocatalysis conditions using an automated microscale multireactor workstation: Synthesis of *meso*-tetramesitylporphyrin. Organic Process Research & Development 3(1), 28–37 (1999)
[25] Cheng, L., et al.: Accelerating electrolyte discovery for energy storage with high-throughput screening. The Journal of Physical Chemistry Letters 6(2), 283–291 (2015)
[26] Senkan, S.M.: High-throughput screening of solid-state catalyst libraries. Nature 394(6691), 350–353 (1998)
[27] Macarron, et al.: Impact of high-throughput screening in biomedical research. Nature Reviews Drug Discovery 10(3), 188–195 (2011)

[28] Dave, A., et al.: Autonomous discovery of battery electrolytes with robotic experimentation and machine learning. Cell Reports Physical Science 1(12), 100264 (2020)
[29] MacLeod, B.P., et al.: Self-driving laboratory for accelerated discovery of thin-film materials. Science Advances 6(20), eaaz8867 (2020)
[30] Shimizu, R., et al.: Autonomous materials synthesis by machine learning and robotics. APL Materials 8(11), 111110 (2020)
[31] Montoya, J.H., et al.: Autonomous intelligent agents for accelerated materials discovery. Chemical Science 11, 8517–8532 (2020)
[32] Pendleton, I.M., et al.: Experiment specification, capture and laboratory automation technology (escalate): A software pipeline for automated chemical experimentation and data management. MRS Communications 9(3), 846–859 (2019)
[33] Noack, M.M., et al.: A kriging-based approach to autonomous experimentation with applications to x-ray scattering. Scientific Reports 9(1), 1–19 (2019)
[34] Flores-Leonar, M.M., et al.: Materials acceleration platforms: On the way to autonomous experimentation. Current Opinion in Green and Sustainable Chemistry 25, 100370 (2020)
[35] Cortes-Borda, D., et al.: An autonomous self-optimizing flow reactor for the synthesis of natural product carpanone. The Journal of Organic Chemistry 83(23), 14286–14299 (2018)
[36] Boyce, B.L., Uchic, M.D.: Progress toward autonomous experimental systems for alloy development. MRS Bulletin 44(4), 273–280 (2019)
[37] Roch, L.M., et al.: Chemos: Orchestrating autonomous experimentation. Science Robotics 3(19), eaat5559 (2018)
[38] Gromski, P.S., Granda, J.M., Cronin, J.: Universal chemical synthesis and discovery with the chemputer. Trends in Chemistry 2(1), 4–12 (2020)
[39] Steiner, S., et al.: Organic synthesis in a modular robotic system driven by a chemical programming language. Science 363(6423), eaav2211 (2019)
[40] Jayabal, H., Dingari, N.N., Rai, B.: A linear viscoelastic model to understand skin mechanical behaviour and for cosmetic formulation design. International Journal of Cosmetic Science 41(3), 292–299 (2019)
[41] Gupta, R., Sridhar, D. B., Rai, B.: Molecular dynamics simulation study of permeation of molecules through skin lipid bilayer. The Journal of Physical Chemistry B 120(34), 8987–8996 (2016)
[42] Gajula, K., et al.: In-silico skin model: A multi-scale simulation study of drug transport. Journal of Chemical Information and Modelling 57(8), 2027–2034 (2017)
[43] Gupta, R., et al.: Effect of chemical permeation enhancers on skin permeability: In silico screening using molecular dynamics simulations. Scientific Reports 9(1), 1456 (2019)
[44] Flick, E.W.: Cosmetic and toiletry formulations. Available at: https://www.sciencedirect.com/book/9780815513063/cosmetic-and-toiletry-formulations#book-info. Accessed 20 May 2021
[45] Flick, E.W.: Industrial water-based paint formulations. Available at: https://www.sciencedirect.com/science/article/pii/B9780815513452500204?via%3Dihub. Accessed 20 May 2021
[46] Wibowo, C., Ng, K.M.: Product-oriented process synthesis and development: Creams and pastes. AIChE Journal 47(12), 2746–2767 (2001)
[47] Wibowo, C., Ng, K.M.: Product-centered processing: Manufacture of chemical-based consumer products. AIChE Journal 48(6), 1212–1230 (2002)

[48] Mori, S., et al.: Flow graph corpus from recipe texts. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), pp. 2370–2377 (2014)
[49] Maeta, H., Sasada, T., Mori, S.: A framework for procedural text understanding. In: Proceedings of the 14th International Conference on Parsing Technologies, pp. 50–60 (2015)
[50] Jermsurawong, J., Habash, N.: Predicting the structure of cooking recipes. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 781–786 (2015)
[51] Kiddon, C., et al.: Mise en place: Unsupervised interpretation of instructional recipes. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 982–992 (2015)
[52] Mysore, S., et al.: Automatically extracting action graphs from materials science synthesis procedures. arXiv preprint arXiv:1711.06872 (2017)
[53] Matsumoto, S., Yamanaka, R., Chiba, H.: Mapping RDF graphs to property graphs. arXiv preprint arXiv:1812.01801 (2018)
[54] Alocci, D., et al.: Property graph vs RDF triple store: A comparison on glycan substructure search. PloS ONE 10(12), e0144578 (2015)
[55] Baken, N.: Linked data for smart homes: Comparing RDF and labeled property graphs. In: LDAC2020—8th Linked Data in Architecture and Construction Workshop, pp. 23–36 (2020)
[56] Kim, S., et al.: Pubchem substance and compound databases. Nucleic Acids Research 44(D1), D1202–D1213 (2016)
[57] Harashima, J., Hiramatsu, M.: Cookpad parsed corpus: Linguistic annotations of Japanese recipes. In: Proceedings of the 14th Linguistic Annotation Workshop, pp. 87–92 (2020)
[58] Majumder, B.P., et al.: Generating personalized recipes from historical user preferences. arXiv preprint arXiv:1909.00105 (2019)
[59] Bergstra, J., et al.: Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems 24, 2546–2554 (2011)
[60] Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. The Journal of Machine Learning Research 13(1), 281–305 (2012)
[61] Anderson, M.J., Whitcomb, P.J.: DOE simplified: Practical tools for effective experimentation. CRC Press, Boca Raton (2017)
[62] Lindauer, M., et al.: Smac v3: Algorithm configuration in Python. Available at: https://github.com/automl/SMAC3. Accessed 20 May 2021
[63] Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems 25, 2951–2959 (2012)
[64] Häse, F., et al.: Phoenics: A universal deep Bayesian optimizer. arXiv preprint arXiv:1801.01469 (2018)
[65] Karande, P., et al.: Design principles of chemical penetration enhancers for transdermal drug delivery. In: Proceedings of the National Academy of Sciences 102(13), 4688–4693(2005)
[66] Eichner, A., et al.: Influence of the penetration enhancer isopropyl myristate on stratum corneum lipid model membranes revealed by neutron diffraction and 2h nmr experiments. Biochimica et Biophysica Acta (BBA)-Biomembranes 1859(5), 745–755 (2017)
[67] Nichols, M.E.: Paint weathering tests. In: Handbook of Environmental Degradation of Materials, pp. 51–67. Elsevier, Amsterdam (2018)
[68] Akafuah, N.K., et al.: Evolution of the automotive body coating process a review. Coatings 6(2), 24 (2016)

[69] Nichols, M., et al.: An improved accelerated weathering protocol to anticipate florida exposure behavior of coatings. Journal of Coatings Technology and Research 10(2), 153–173 (2013)
[70] Makki, H., et al.: A simulation approach to study photo-degradation processes of polymeric coatings. Polymer Degradation and Stability 105, 68–79 (2014)
[71] Hinderliter, B., Croll, S.: Monte Carlo approach to estimating the photo-degradation of polymer coatings. Journal of Coatings Technology and Research 2(6), 483–491 (2005)
[72] Schichtel, J.J., Chattopadhyay, A.: Modelling thermoset polymers using an improved molecular dynamics crosslinking methodology. Computational Materials Science 174, 109469 (2020)
[73] Chaube, S., et al.: Multiscale analysis of large-strain deformation behaviour of random crosslinked elastomers. Molecular Simulation 45(2), 111–119 (2019)

AUTHOR BIOGRAPHY

Dr. **Sagar Sunkle** is a senior scientist at Tata Consultancy Services Research, India. Sagar received his PhD (with distinction) in Software Engineering from Otto-von-Guericke University (OvGU) of Magdeburg, Germany. He also has two masters, one in Computer Science with specialization in soft computing technologies from the University of Mumbai, India and second in Data and Knowledge Engineering from OvGU, both also with distinction. Sagar has over 40 publications including conferences, journals, book chapters and patents. His current research interests include using natural language processing and machine learning to derive insights from information and transform insights into recommendations with applications to material informatics, banking and financial services, and related business domains.
ORCID: 0000-0002-4757-6256

Deepak Jain received his Master's in Thermal and Fluid Engineering from Indian Institute of Technology Bombay (IIT Bombay) in 2015. Soon after his Master's, he joined the Corporate Technology Office of Tata Consultancy Services as a junior researcher in the Physical Sciences Group. His research interests include computational fluid dynamics, applied machine learning to physical sciences, 3rd generation photovoltaic materials and molecular modelling. Deepak has filed 4 patents and has over 8 publications including journals, conferences and a book chapter.
ORCID: 0000-0001-5849-454X

Krati Saxena is a Scientist at Tata Research Development and Design Center, Pune, India (TRDDC). Previously, she completed a Bachelor's degree in System Science from Indian Institute of Technology Jodhpur and received a Master's degree in Global Advanced Assistive Robotics course from Kyushu Institute of Technology, Japan. Her current research interests include content mining, natural language processing, textual AI, data-driven decision-making, and knowledge discovery.
ORCID: 0000-0001-7049-9685

Ashwini Patil is a researcher at Tata Research Development and Design Centre, Pune, India (TRDDC). She is working in the domain of Natural Language Processing. She has completed a Master of Technology in Computer Science from Visvesvaraya National Institute of Technology, Nagpur, India (VNIT Nagpur, India). Her current research interests include graph databases, semantic similarly, and deep learning for information and knowledge discovery.
ORCID: 0000-0002-7948-0427

Tushita Singh is a Researcher at Tata Research Development and Design Centre, Pune, India (TRDDC). She is a graduate in Computer Science from the Indian Institute of Information Technology (IIIT), Nagpur, India. Her current areas of research interests include natural language processing, data-driven decision making, and textual AI.
ORCID: 0000-0002-1288-8161

Dr. **Beena Rai** is the head of the Physical Sciences Research Area with over 20 years of experience at Tata Consultancy Services Research, India. Beena is a PhD from India's premier research institution—National Chemical Laboratory, Pune. Her research focuses on the molecular modelling based rational design and development of surfactants for various industrial applications such as mineral processing, ceramics, paints and coatings, cements, lubricants, agrochemicals, pharmaceuticals and cosmetics. Beena has published close to 100 research papers in reputed journals and conferences. She has 22 patent/patent applications to her credit. She has delivered several keynotes and invited talks at various forums. She is the editor of the book titled *Molecular Modelling for the Design of Novel Performance Chemicals and Materials*, published by Taylor & Francis, CRC Press. Beena is a recipient of the prestigious Chevening Scholarship at Said Business School, University of Oxford, UK, and several other recognitions and awards.

ORCID: 0000-0002-8637-7778

Vinay Kulkarni is a Chief Scientist and Head of Software Systems Research at Tata Consultancy Services, India. His research interests include model-driven software engineering, enterprise modelling and software engineering for an uncertain world. His work in model-driven software engineering has led to a tool set that has been used to deliver several large business-critical systems over the past 20 years. Much of this work has found a way into OMG standards. This work also received fair mention in respected international print media. He has several patents to his credit and has authored more than 100 papers in scholastic journals and conferences worldwide. He has served as the conference and program chairperson for the premier ACM and IEEE international conferences in the area of software engineering, and is on technical program committees of many international conferences. Recently, he was inducted as Fellow of Indian National Academy of Engineering. An alumnus of Indian Institute of Technology Madras, Vinay also serves as Visiting Professor at Middlesex University, London.

ORCID: 0000-0003-1570-1339